PRIVILEGED AND CONFIDENTIAL (PLEASE DO NOT SHARE)

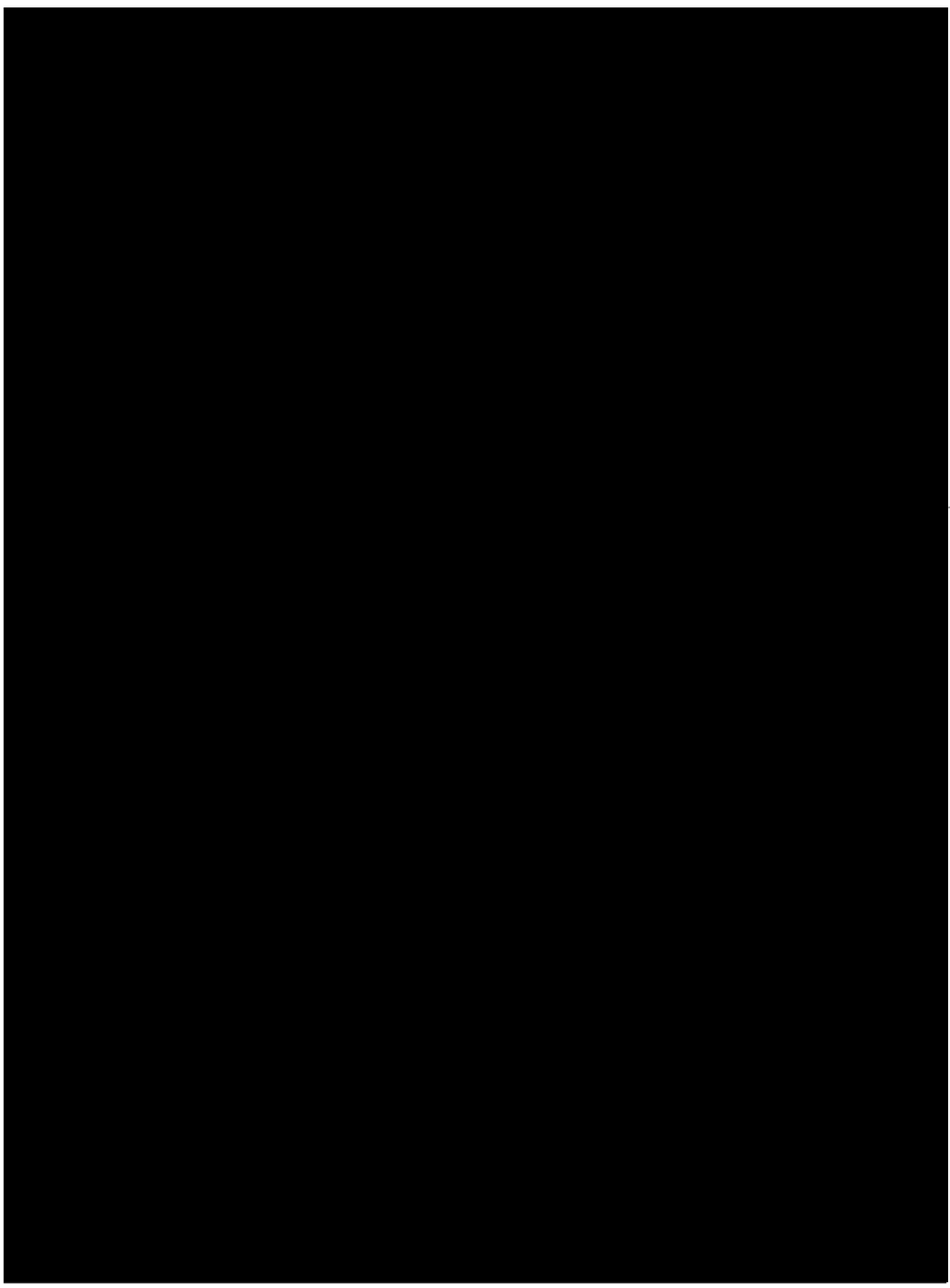
Status: Final

Authors: haou@google.com, tlipus@google.com, gTrade team

Last Updated: 2019-09-03

go/gdn-adx-first-price-bidding-infra

Adx will launch its first price auction in 2019. For GDN advertisers, in order to bid into this Adx first price auction, the first price auction eligibility signals need to be piped through the buyside infrastructure. Serving infrastructure also has to be changed in order to support serving the future bidding models for GDN advertisers when bidding into Adx first price auction. This document describes the proposed design in order to make Adx first price bidding available to GDN advertisers.



Comment [1]: [redacted]

Comment [2]: Done.

Comment [3]: [redacted]

Comment [4]: [redacted]

GdnFirstPriceBiddingModel::SetOutput method. The bidding model class should be put under gdn sub-directory here, and the new GdnPredictionCollection and GdnPrediction proto messages should be put in file gdn-predictions.proto under contentads/cat2/proto directory.

GDN query level first price prediction model

If the first price prediction model only use query level features, the prediction for each Ad candidate for this query will be the same. In this case, we do not need to make predictions for each individual Ad candidate. To serve such a query level prediction model for GDN Ads, we can create a similar subclass of Cat2SmartassPredictionModel like the impression level model described in the previous section, and use similar proto fields to store the prediction value in the AdCandidatePredictionInfo proto message so that the prediction values will be successfully passed to the Supermixer and consumed by related first price util classes. The only differences are:

- IsModelQueryOnly() method should return true to avoid making predictions for each Ad candidate.
- ShouldPredictImpression() method should always return true since we don't make predictions for each impression. Instead, when we assign the prediction value to each Ad candidate in the *SetOutput* method, we should only set the prediction value to the Ad candidates that are qualified for the query level model predictions.

It seems that we store replicated information by copying prediction values from query level smartass model into each Ad candidate. The ideal approach should be finding some place to store and pass the query level prediction to the infrastructure where the prediction value is consumed. The reasons we go with the current design are:

- Store the smartass model prediction as query level requires a complete new design of changes to the infrastructure, including changes in Cat2 Mixer and Supermixer. This requires a lot of work. Given the tight schedule of first price project, we choose the current solution which is a reasonable design that requires minimum changes to infrastructure so that we can start experiments in time.
- In the future, when Adx first price traffic ramps up, we may launch improved first price prediction model with Ad dependent features. In this case, the infrastructure that related to query level model we build today may turn out to be a waste of time. Current design can handle both query level and impression level prediction model serving reasonably well.

We will keep track of this issue as Adx/AdMob first price traffic ramping up. If we see the needs of building infrastructure for serving and logging query level prediction models in Supermixer and Cat2 mixer, we will develop it in the future.

Passing GDN first price predictions to Supermixer

The predictions we get from Smartass model server are placed in AdCandidate.csm.contentads_data.gdn_prediction_collection field. At this point, we still do not have a clear picture about how the prediction model for GDN first price bidding looks like and whether only query level features or both query and Ad candidate level features will be used in the prediction models. In order to have more flexibility for building GDN prediction models and

Comment [5]: [redacted]

Comment [6]: [redacted]

[redacted]

Comment [7]: [redacted]

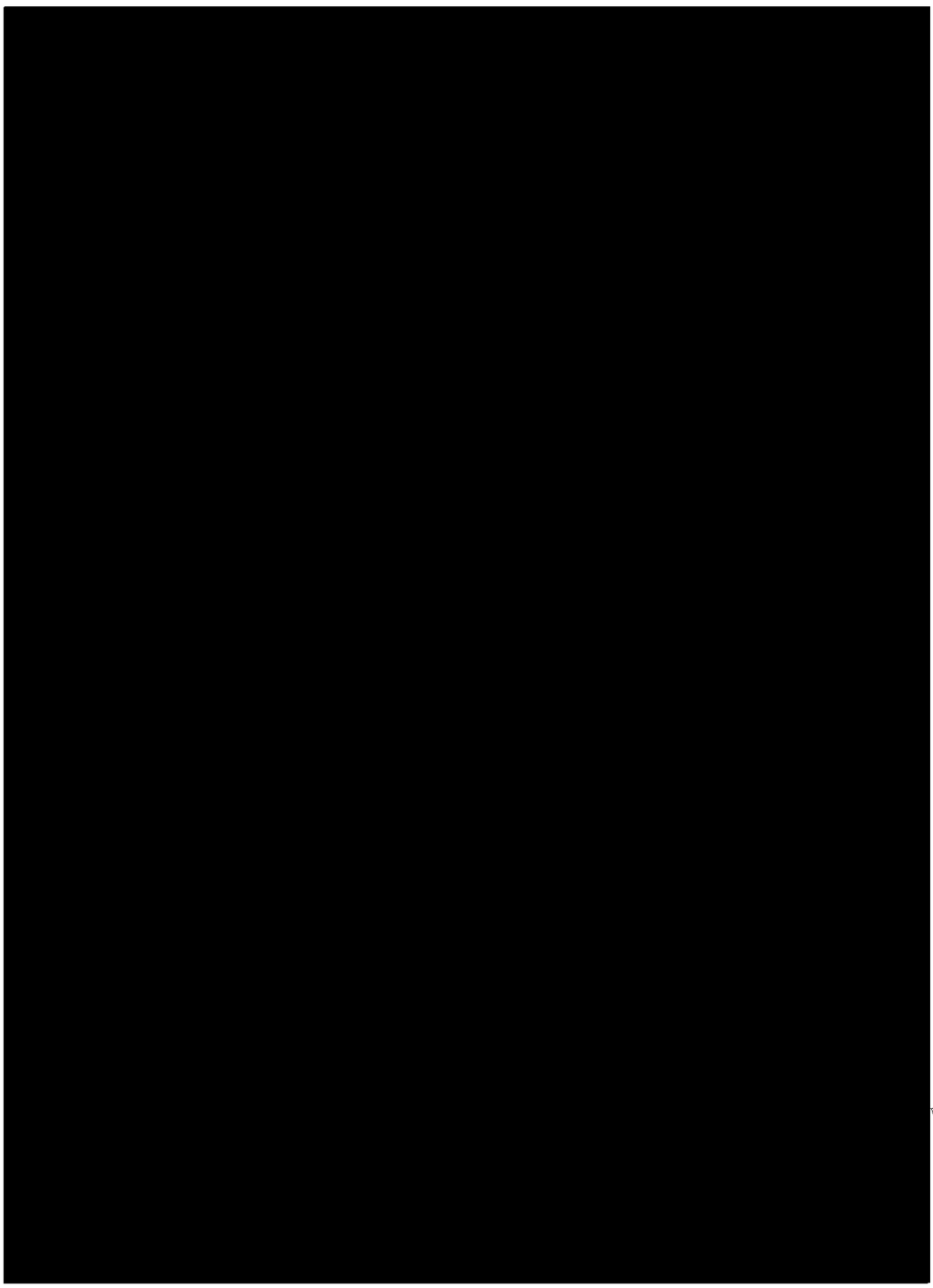
Comment [8]: [redacted]

Comment [9]: [redacted]

Comment [10]: [redacted]

Comment [11]: [redacted]

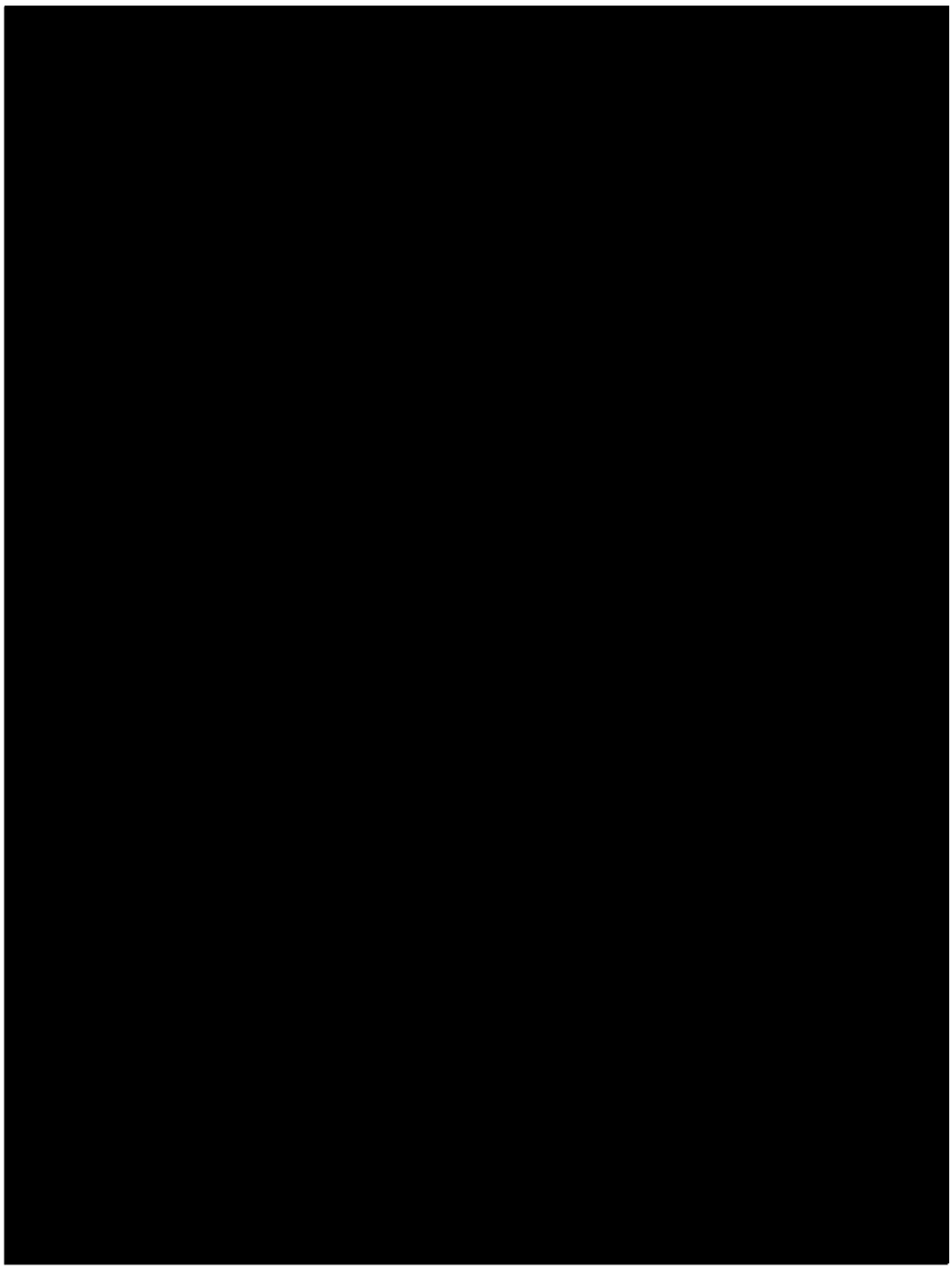
Comment [12]: Done.



Comment [13]:

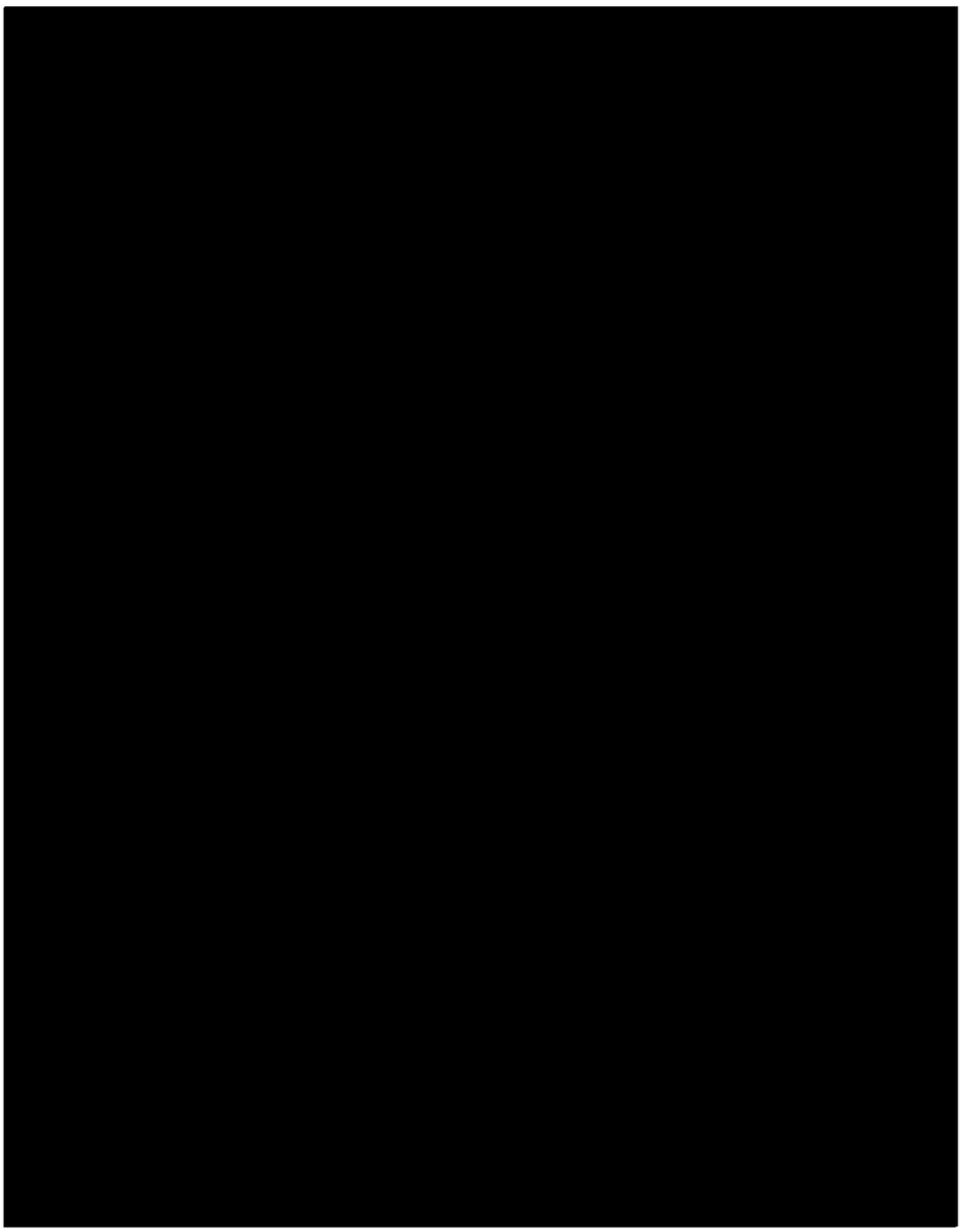
Comment [14]:

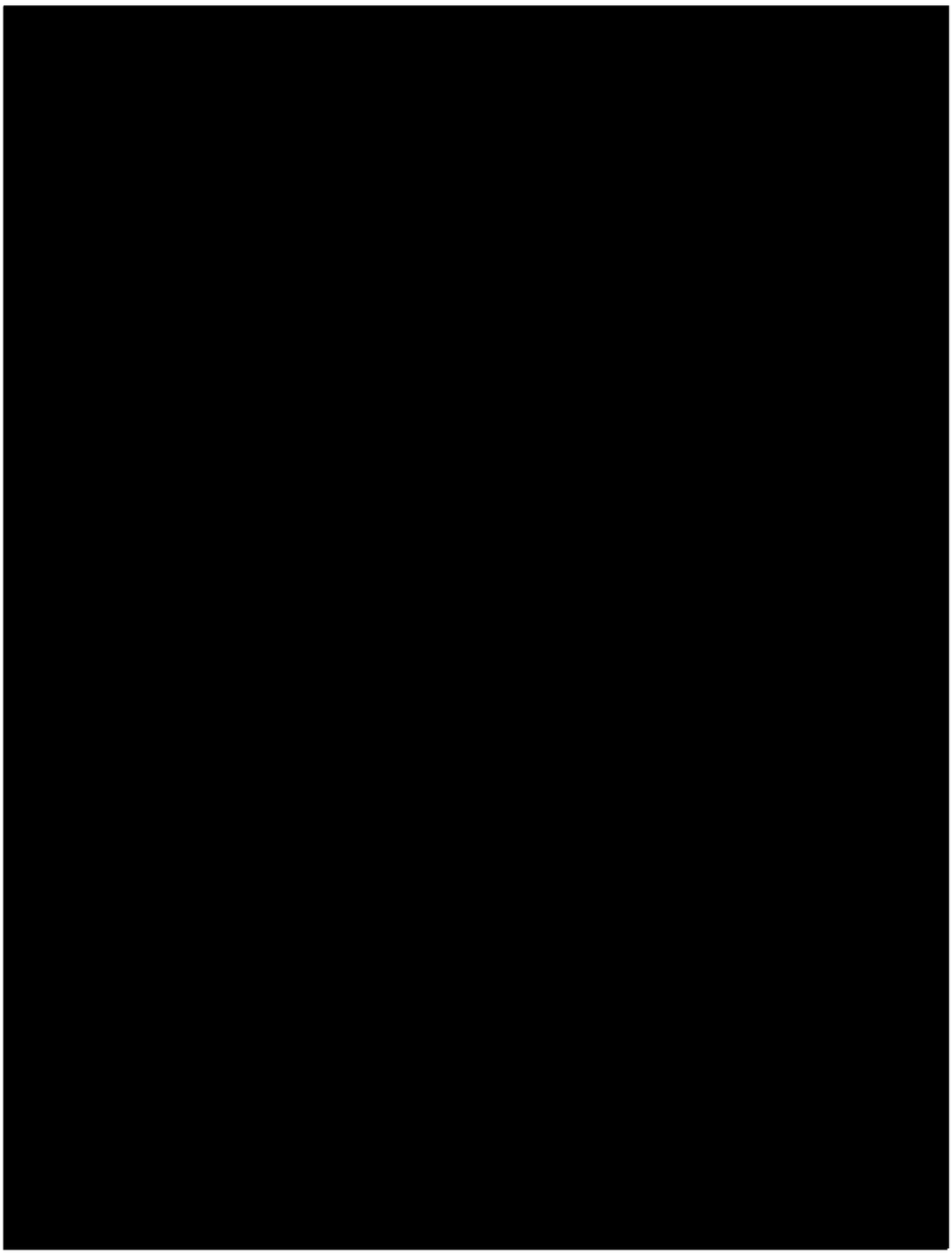
Comment [15]: SGTM



Comment [16]: [redacted]

Comment [17]: Done.





Comprehensive unit tests.

Date	Author	Description	Reviewed by	Signed off by
2019/01/24	haou	Initial draft of Adx first price bidding infra design for GDN advertisers.		
2019/03/26	haou	Add sections to describe how to handle the special case of multi-slot text Ads and update advertiser's cost for Adx first price auction.		
2019/05/28	haou	Add a section to describe serving query level smartass prediction model.		
2019/06/21	haou	Update the cost computation section when first price bernanke is enabled.		
2019/09/03	haou	Update the reserve price aware first price bidding, and cost computation for non-VBB traffic and when reserve price is used in first price bidding.		